Improving Object Detection by Label Assignment Distillation*

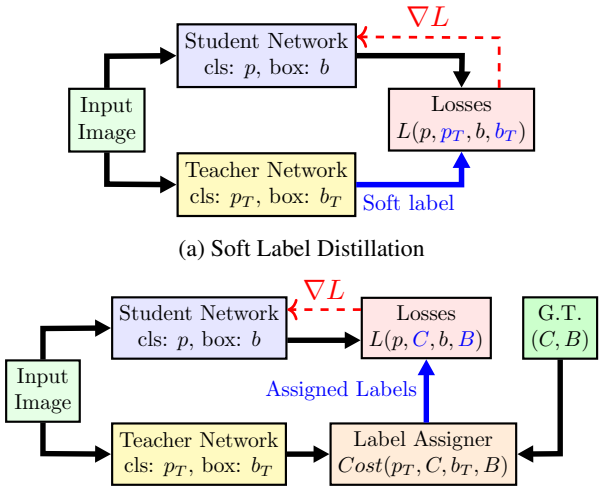
Chuong H. Nguyen†, Thuy C. Nguyen†, Tuan N. Tang, Nam L.H. Phan
CyberCore AI, Ho Chi Minh, Viet Nam
chuong.nguyen,thuy.nguyen,tuan.tang,nam.phan@cybercore.co.jp

Abstract

Label assignment in object detection aims to assign targets, foreground or background, to sampled regions in an image. Unlike labeling for image classification, this problem is not well defined due to the object's bounding box. In this paper, we investigate the problem from a perspective of distillation, hence we call Label Assignment Distillation (LAD). Our initial motivation is very simple, we use a teacher network to generate labels for the student. This can be achieved in two ways: either using the teacher's prediction as the direct targets (soft label), or through the hard labels dynamically assigned by the teacher (LAD). Our experiments reveal that: (i) LAD is more effective than soft-label, but they are complementary. (ii) Using LAD, a smaller teacher can also improve a larger student significantly, while soft-label can't. We then introduce Co-learning LAD, in which two networks simultaneously learn from scratch and the role of teacher and student are dynamically interchanged. Using PAA-ResNet50 as a teacher, our LAD techniques can improve detectors PAA-ResNet101 and PAA-ResNeXt101 to 46AP *and* 47.5AP *on the COCO test-dev set. With a stronger teacher PAA-SwinB, we improve the students PAA-ResNet50 to* 43.7AP *by only* 1× *schedule training and standard setting, and PAA-ResNet101 to* 47.9AP*, significantly surpassing the current methods. Our source code and checkpoints will be released at link.*

1. Introduction

Object detection is a long-standing and fundamental problem, and many algorithms have been proposed to improve the benchmark accuracy. Nevertheless, the principal framework is still unchanged: an image is divided into many small sample regions, each is assigned a target, on which the detector is trained in a supervised manner. Reviewing the literature, a majority are dedicated to inventing new architectures [31, 36, 37, 1, 42, 22, 38, 3] or defining effective training loss functions [28, 5, 26, 35]. However, the most frontal problem of supervised learning, *i.e.*, how to assign the training targets, yet get less attention.



(a) Soft Label Distillation

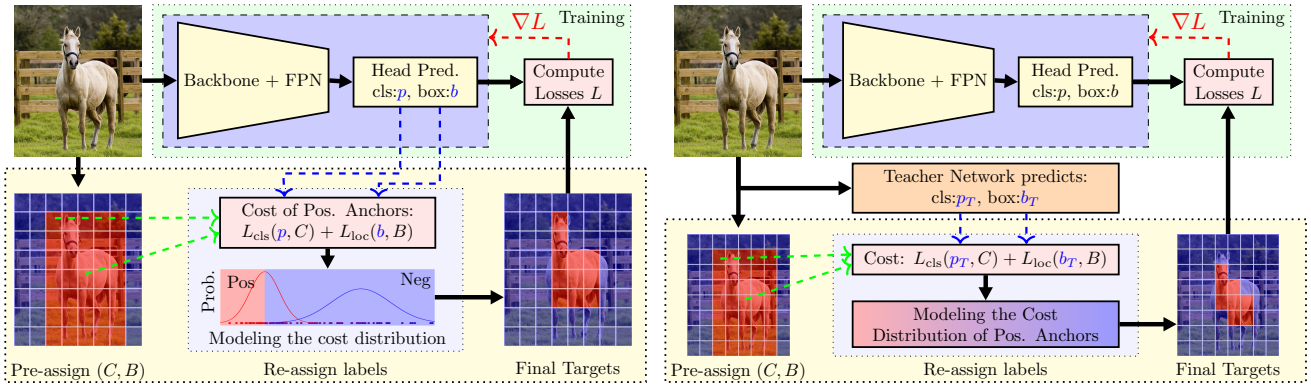
(b) Label Assignment Distillation

Figure 1: Compare Soft Label Distillation and Label Assignment Distillation (LAD). In the former, the teacher's output is directly used as the target, while in the latter, it is used to evaluate the cost for label assignment.

Unlike image classification, where a category can be easily set to an image, defining labels in object detection is ambiguous due to bounding boxes' overlapping. Obviously, if a region is completely overlapping or disjoint with a ground truth box, it is definitely positive (foreground) or negative (background). However, for a partial overlapping case, how should we consider it?

A common practice to define a positive region is if its Intersection over Union (IoU) with the nearest ground truth box at least 0.5. This may due to the high recall preference in the VOC [14] and COCO [29] evaluations. Ironically, a network trained with low IoU assignment yields high recall but noisy predictions, while using only high IoU samples degrades the performance [4]. In addition, regions with the same IoU can have different semantic meaning for classification. Hence, recent research [21, 53] suggests that solely relying on IoU is insufficient, and a combination of classification and localization performance is preferred.

*Paper is under review.
†Equal Contribution.



(a) Probabilistic Anchor Assignment (PAA)

(b) Label Assignment Distillation (LAD)

Figure 2: PAA[21] and its LAD counterpart. PAA uses the prediction at step (t) to compute the label assignment cost for the next step $(t+1)$. It is a type of bootstrap learning, where the network updating and label assignment form a closed-loop system. In contrast, LAD uses an independent teacher, which decouples the label assignment and training processes.

This paper investigates the label assignment problem from the perspective of distillation technique, hence we call Label Assignment Distillation (LAD). Regarding network distillation, the common approach is to directly use the teacher output as the targets, i.e. dark knowledge [19], to train the student as illustrated in Fig. 1a. Note that, for object detection, the classification and localization tasks are often distilled independently, with or without using the ground truth. In contrary, Fig. 1b illustrates our proposed LAD, where the classification and localization prediction of the teacher and the ground truth are fused into the cost function before being indirectly distilled to the student.

Our proposed LAD is inspired from the realization that the label assignment methods [6, 21, 53, 15] are forms of self-distillation [7, 16, 30]. For instance, Fig. 2a illustrates the Probabilistic Anchor Assignment (PAA) [21]. As shown Fig. 1b, from LAD perspective the network itself can be seen as the teacher, that is, the current prediction is used to assign label for the next learning step. In contrary, Fig. 2b illustrates the LAD counterpart, where an independent teacher is used to compute the assignment cost, hence decoupling the training and label assignment processes.

Motivated by this observation, we conduct a pilot study about LAD's properties. Consequently, we propose the Co-learning LAD, in which both networks can be trained from scratch, and the roles of teacher and student are dynamically interchanged. Our contributions are summarized as:

- We introduce the general concept of Label Assignment Distillation (LAD). LAD is very simple but effective, applicable to many label assignment methods, and can complement other distillation techniques. More important, using LAD, a smaller teacher can improve its student significantly, possibly surpasses the teacher by a large margin.
- We propose the Co-learning Label Assignment Distillation (CoLAD) to train both student and teacher simultaneously. We show that two networks trained with CoLAD are significantly better than if each was trained individually, given the same initialization.
- We achieve state-of-the-art performance on the MS COCO benchmark. Using PAA-ResNet50 as a teacher, CoLAD improves PAA-ResNet101 to $46.0\text{AP}(+1.2)$, and PAA-ResNeXt101 to $47.5\text{AP}(+0.9)$. With Swin-B Transformer backbone, PAA detector trained with LAD achieves 51.4AP on *val* set, approaching the Cascade Mask-RCNN (51.9AP) [32], without the costly segmentation training. Remarkably, the results are all achieved simply with smaller teachers. Finally, recycling the PAA-Swin-B as a teacher, we improve PAA-ResNet50 to 43.7AP (with $1\times$ schedule training), and PAA-ResNet101 to 47.9AP.

2. Related Work

2.1. Label Assignment in Object Detection

Modern object detectors can be single or multi-stages. Single-stage is simpler and more efficient, while multi-stages are more complex but predict with higher precision.

In **single stage detectors**, classification and localization are predicted concurrently for each pixel. There are anchor-based or anchor-free methods. For anchor-based detectors, such as SSD [31], RetinaNet [28], an anchor is typically considered as positive if its IoU with the nearest ground truth box is at least 0.5, negative if less than 0.4, and ignored otherwise. ATSS [48] improves RetinaNet by selecting 9 nearest anchors from the object center in each feature level, and use the mean plus standard deviation (std) of IoU from this set as the assignment threshold. Anchor-free methods

Table 1: Comparing hard assignment methods in three aspects: pre-assignment, cost evaluation, and re-assignment.

Methods	Pre-assignment	Cost Evaluation	Re-assignment
ATSS [48]	Top-k points closest to the object's center	Anchor IoU	Modeling the scores by a Gaussian. Select the mean plus standard deviation as threshold.
MAL [20]	Anchor IoU >0.5	Classification and Localization loss	All-to-top 1. The number of positive anchors gradually reduces from all (first iters) to top-1 (last iters).
PAA[21]	Anchor IoU >0.1	Classification and Localization loss	Modeling the scores by a mixture of two Gaussian. Select center of the Gaussian with lower mean as threshold.
DETR [6]	All points	Classification and Localization loss	Modeling the scores as an optimal transportation cost. Using Hungarian assignment to select top-1.
OTA [15]	Top-r^2 points closest to the object's center	Cls., Loc. loss and center prior	Modeling the scores as an optimal transportation cost. Using Sinkhorn-Knopp Iteration to select top-k.

instead resort objects by points. FCOS [42] assigns a point as positive if it is in a ground truth box, negative otherwise. If the point falls into two or more ground truth boxes, the smallest box is chosen. CornerNet [22] represents an object by two corners of the bounding box, and only the corner points are assigned as positive. During training, the farther the points from the corner center, the larger weight it contributes to the loss function as negative samples.

In **multi-stage detectors**, the first stage network proposes candidate regions, by which the features are cropped and fed to the following stages for the box refinement and class prediction. The process can continue in a cascade manner as many stages as needed. Anchor assignments are also different in each stage. In the first stage, *i.e*. proposal network, an anchor is typically considered as positive if its IoU with any ground-truth is greater than 0.7, negative if less than 0.3, and ignored otherwise. For the second stage, 0.5 IoU threshold is used to separate positive and negative anchors. Cascade R-CNN [3] improves Faster R-CNN by adding more stages. Setting the IoU threshold too high leads to extremely few positive samples, thus insufficient to train the network. Therefore, Cascade R-CNN re-assign the samples with IoU thresholds progressively increasing from 0.5 to 0.7 after each stage.

In summary, label assignments in the aforementioned methods are static and heuristically designed.

2.2. Learning Label Assignment

Learning label assignment approaches update the network weights and learn the assignment iteratively towards final optimization. There are two main approaches.

In **hard assignment**, samples can only be either positive or negative. The pipeline generally includes three steps. We first select and *pre-assign* a set of potential samples as positive, then *evaluate* their performance by a cost function. Finally, we select a threshold to separate the samples having lower cost as true positive, while the rest are *reassigned* to negative or ignored. Table 1 compares different methods of learning label assignment in these aspects for single stage detector. For two-stage detectors, Dynamic-RCNN [47] progressively increases the IoU threshold in the second stage by mean of the predicted IoU of the proposal network, and modifies the hyper-parameter of SmoothL1 Loss function according to the statistics of regression loss.

In **soft assignment**, samples can have both positive and negative semantic characteristics. There are two main approaches: soft-weighted loss and soft-targets supervision. In the former, a set of positive candidates are first selected by a relaxed criterion, *e.g*. low IoU threshold, then their importance scores are evaluated and used as weights to sum the loss values of the samples. FreeAnchor [50] first constructs a bag of top-k anchors having highest IoU with object, then calculates the importance scores from the classification and localization confidence using Mean-Max function. Following this framework, NoisyAnchor [23] derives the importance scores, *i.e*. cleanliness, from the regression and classification losses of the samples. SAPD [54] utilizes distance to object center as the metrics to measure the importance of samples. Furthermore, it leverages FSAF [55] module to assign soft-weights to samples in different feature levels. Alternatively, soft-target supervision decides how much a sample belonging to positive or negative based on an estimated quality distribution. AutoAssign [53] improves FCOS [42] by estimating object's spatial distribution, *e.g*. a 2D-Gaussian, for each category. Then, a point in the ground truth box is supervised by both positive and negative losses weighted by this distribution. Similarly, IQDet [33] builds a mixture of Gaussian models to capture IoU quality. Differently, positive points are randomly sampled by this distribution and via a bi-linear interpolation. The quality distribution is also used as the soft-label for classification supervision.

2.3. Distillation in Object Detection

Distillation is first introduced in the seminal work of Hilton *et al*. [19] for image classification. By using the prediction of a teacher network as soft training targets, we can distill the knowledge from the teacher to a student. Distilla-

tion then becomes the principle for many other related problems, such as self-supervised learning. However, applying distillation to object detection is not straightforward, since classification and localization must be learned simultaneously. Zheng *et al*. [51] hypothesize that a teacher is better in estimating the localization uncertainty, which can be used as the dark knowledge for distillation. However, the method is only applicable to Generalized Focal detector [26] due to its formulation problem. Other works focus on distillation by feature mimicking. Chen *et al*. [8] combine feature mimicking and soft-label distillation on Faster-RCNN. Other researches [46, 43, 24] believe that background features are noisy, thus propose applying semantic masks to focus attention on the meaningful regions, especially in and near foregrounds. In contrast, Guo *et al*. [17] consider that background can capture the object's relationship and suggest decoupling the background features during distillation.

Although these methods exploit different ways for distillation, they all enforce a direct mimicking the teacher's final or intermediate outputs. This may be too restrictive if their architectures or performance are significantly different. Therefore, several papers [34, 51] propose to add a teacher assistant network to bridge their gaps. Nevertheless, this introduces more hyperparameters, larger memory, longer training, and complicates the process.

Our proposed LAD extends the distillation concept and is orthogonal with these methods. Generally, they can be combined to further improve the performance.

3. Method

3.1. Soft-label distillation

In network distillation, the teacher's prediction is referred as soft-label, which is believed to capture the network's dark knowledge and provide richer information for training a student. Adopting the convention from the classification problem [19], we use the Kullback-Leibler (KL) divergence loss. Follow Focal loss [28], we also add the focal term to deal with the extreme imbalance problem

$$KL(p_t, p_s) = \sum_{c=1}^{C} w^c \left(p_t^c \log \frac{p_t^c}{p_s^c} + (1 - p_t^c) \log \frac{1 - p_t^c}{1 - p_s^c} \right), \tag{1}$$

where p_t and p_s denote the teacher and student prediction (i.e. after Sigmoid), c is the class index of total C classes in the dataset, and $w^c = |p_t^c - p_s^c|^\gamma$ is the focal term. Note, when replacing the soft-label p_t with an one-hot vector, (1) becomes the standard Focal loss. Alternatively, we can also use $\mathcal{L}_1$ or $\mathcal{L}_2$ losses. Different from the classification losses, localization loss is only computed for positive samples. Hence, we select the predicted boxes having $IoU > 0.5$ w.r.t. its nearest ground truth box for localization distilling.

3.2. Label Assignment Distillation

LAD is very general and can be applied to many learning label assignment methods, such as those described in Sec. 2.2. For a concrete example, we adopt PAA to implement LAD. In PAA, for each object, we select a set of anchors $\{a_i\}$ that have $IoU \geq 0.1$, and evaluate the assignment cost

$$c_i = FL(p_i, C) + (1 - IoU(b_i, B)) \tag{2}$$

as if they are positive anchors, where FL is the Focal loss, (p_i, b_i) are the predicted class probability and bounding box of a_i, (C, B) are the ground truth class and bounding box of the object. Then, we build a distribution model for $\{c_i\}$ by a mixture of two Gaussian: $G(\mu_1, \sigma_1) + G(\mu_2, \sigma_2), \mu_1 < \mu_2$. We then assign anchors with $c_i < \mu_1$ as the true positive, and reassign the other as negative, as illustrated in Fig. 2a.

To convert PAA to LAD, we simply use the teacher's prediction (p_T^i, b_T^i) instead of (p_i, b_i) in (2), and proceed exactly the steps, as illustrated in Fig. 2b. Consequently, LAD presents a new distillation method for object detection that works without directly mimicking the teacher's outputs.

3.3. A Pilot Study

Our motivation is to seek a method that utilizes the knowledge from teacher network, and this can be achieved either by soft-label or LAD, as shown in Fig. 1. Therefore, we conduct a pilot study to understand their properties.

3.3.1 A preliminary comparision

We first compare the two methods using PAA detector with ResNet backbone R101 and R50 as teacher and student, and report the results for the student PAA-R50 in Tab. 2.

Table 2: Compare the performance of the student PAA-R50 using Soft-Label, Label Assignment Distillation (LAD) and their combination (SoLAD) on COCO validation set. (*) denotes longer warming-up learning rate (3000 iterations).

Method	γ	mAP	Improve
Baseline PAA	2	40.4	–
Soft-Label - KL loss*	0	39.8	−0.6
Soft-Label - KL loss*	0.5	41.3	+0.9
Soft-Label - KL loss	2.0	39.6	−0.8
Soft-Label - $\mathcal{L}_1$ loss*	–	40.4	0.0
Soft-Label - $\mathcal{L}_2$ loss	–	41.0	+0.6
LAD (ours)	2	**41.6**	**+1.2**
SoLAD (ours) - KL loss	0.5	**42.4**	**+2.0**

Table 2 shows that both methods can improve the baseline. For soft-label distillation, KL loss is better than $\mathcal{L}_1$ and $\mathcal{L}_2$ losses, but it must be tuned carefully. We found that

a small positive $\gamma = 0.5$ is critical to achieve good performance (41.3AP). Notably, this makes the training unstable due to large initial error, which hence requires a much longer warming-up learning rate. In contrast, LAD yields the higher result (41.6AP) with the standard setting.

Finally, we show that LAD can be easily combined with soft-label distillation. Concretely, label assignment and training losses are conducted exactly as in LAD, but with additional soft-label distillation losses. We name this combination as **SoLAD**. As shown in Tab. 2, SoLAD can improve the baseline PAA by +2.0AP, which accumulates the improvement from each component: +0.9AP of soft label and +1.2AP of LAD. Remarkably, the student PAA-R50's performance (42.4AP) closely converges to its teacher PAA-R101 (42.6AP).

The results above are very promising. It shows that LAD is simple but effective, orthogonal but complementary to other distillation techniques, and worth further exploration.

3.3.2 Does LAD need a bigger teacher network?

Conventionally, a teacher is supposed to be the larger and better performance network. To verify this assumption, we conduct a similar experiment as in Sec. 3.3.1, but swap the teacher and student. The results are shown in Tab. 3.

Table 3: Compare Soft-Label and Label Assignment Distillation (LAD) on COCO validation set. Teacher and student use ResNet50 and ResNet101 backbone, respectively. 2× denotes the 2× training schedule.

Method	Teacher	Network	mAP	Improve
Baseline	None	PAA-R101	42.6	–
Baseline (2×)	None	PAA-R101	43.5	+0.9
Soft-Label	PAA-R50	PAA-R101	40.4	−2.2
LAD (ours)	PAA-R50	PAA-R101	**43.3**	**+0.7**

Table 3 demonstrates a distinctive advantage of LAD. Specifically, using the teacher PAA-R50 (40.4AP) with soft-label distillation, the student PAA-R101's performance drops by −2.2AP relative to the baseline. In contrast, LAD improves the student by +0.7AP and achieves 43.3AP, asymptotically reaching the baseline trained with 2× schedule (43.5AP). Especially, the student surpasses its teacher with a large margin (+2.9AP). This proves a very useful and unique property of LAD - a bidirectional enhancement, because potentially any teacher now can improve its student regardless of their performance's gap.

3.4. Co-learning Label Assignment Distillation

The pilot study in Sec. 3.3 proves that the proposed LAD is a simple and effective solution to improve network training for object detection. However, like many other distillation methods, it requires a teacher pretrained in advance. Learning label assignment methods [6, 21, 53, 15] on the other hand, use a bootstrap mechanism to self-distill the labels. However, a potential drawback is that it can be trapped in local minima. This is because it only exploits the highest short-term reward, i.e. lowest assignment cost, without any chance for exploration.

Therefore, to avoid these drawbacks and combine their advantages, we propose the Co-learning Label Assignment Distillation (CoLAD). CoLAD does not need a beforehand teacher, while potentially avoiding falling into local minima. The framework is illustrated in Fig. 3. Concretely, we use two separate networks similarly to LAD, but none of them must be pre-trained. Instead, we train them concurrently, and dynamically switch the role of teacher and student based on their performance indicator ρ. We propose two criteria for ρ, namely Std/Mean score ($\rho_{\sigma/\mu}$) and Fisher score (ρ_{Fisher}). Let $\{c_i\}$ be the set of the assignment costs evaluated by a network on the pre-assigned positive anchors, the criteria are respectively defined as:

$$\rho_{\sigma/\mu} = \frac{\sigma}{\mu}, \qquad \rho_{\text{Fisher}} = \frac{(\mu^+ - \mu^-)^2}{\sigma^{+2} + \sigma^{-2}}, \tag{3}$$

where (μ, σ) are the mean and std. of $\{c_i\}$, and in the Fisher Score, we approximate the distribution of $\{c_i\}$ by a mixture of two Gaussian models $G(\mu^+, \sigma^+) + G(\mu^-, \sigma^-)$.

At each iteration, the network with higher ρ can be selected as the teacher. Therefore, by dynamically switching the teacher-student, the process no longer relies on a single network, stochastically breaking the loop of training-self assignment. This also brings more randomness at the initial training due to network initialization, hence potentially reducing the risk of local minimal trapping.

Finally, we can also apply the dynamic switch (3) to LAD, in which the teacher is pretrained and fixed. This allows the student learn from the teacher at the early training stage. However, when the student is gradually improved and potentially surpasses its teacher, it can switch to the self-distillation mode. This makes LAD more adaptive.

4. Experiments

We evaluate the proposed methods on the MS COCO benchmark [29], which has about 118k images in the *train* set, 5k in the *val* set and 20k in the *test-dev* set. Following the common protocol, we report the ablation studies on the *val* set, and compare with previous methods on the *test-dev* set using the official COCO evaluation tool.

4.1. Implementation Details

We use PAA detector with different ResNet backbones R18, R50, R101 [18] with FPN [27] for the baseline. The backbones are initialized with weights pre-trained on ImageNet [12]. Our code is implemented using MMDetection

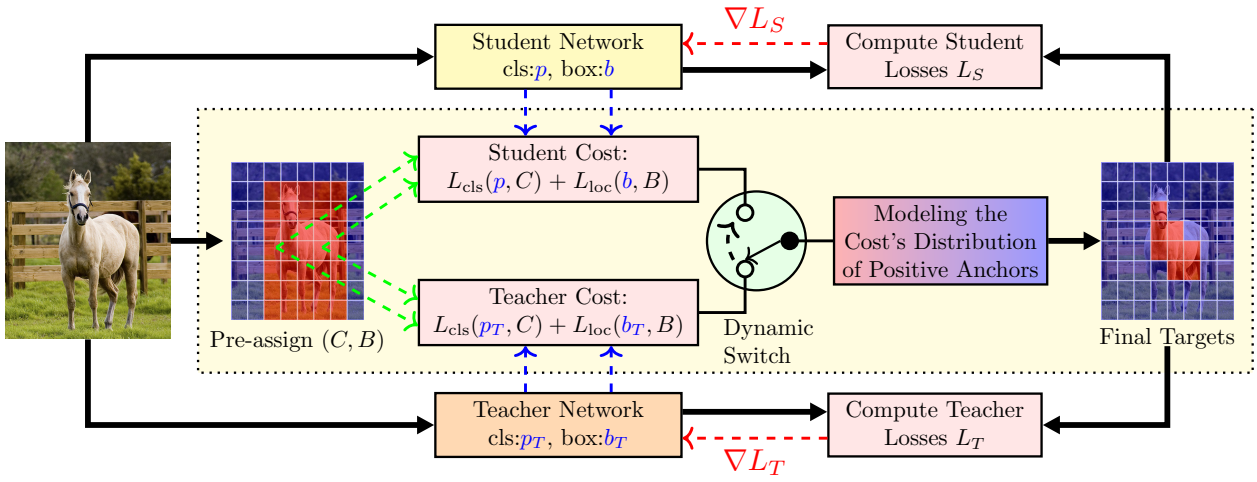


Figure 3: Co-Learning Label Assigment Distillation (CoLAD) framework. Dynamic Switch is described in (3).

framework [9], and follows its standard setup. Specifically, we use SGD optimizer with weight decay of 0.0001 and the momentum of 0.9. $1\times$ training schedule includes 12 epochs with initial learning rate of 0.01, which is then divided by 10 at epoch 8 and epoch 11. $2\times$ training schedule includes 24 epochs with initial learning rate of 0.01, which is then divided by 10 at epoch 16 and epoch 22. Random horizontal flipping is used in data augmentation. We use batch size 16, with image size of (800×1333) pixels for training and testing. All other hyperparameters, such as loss weights and post-processing, are identical to the PAA baseline.

4.2. Ablation Study for CoLAD

In CoLAD, the roles of teacher-student are not persistent, so the terminology teacher and student networks may not exactly apply. However, for convenience, we still call the network with higher performance at initial as the teacher. If two networks are not pretrained, then the one having more parameters and layers is supposed to be the initial teacher.

4.2.1 Switching Criteria in CoLAD

First, we want to know which switching criteria, the Std/Mean or Fisher score, is more suitable for CoLAD, and how CoLAD performs in comparison with the baseline PAA and LAD. Since LAD uses teacher PAA-R50 pretrained with $1\times$ schedule, we also adopt it in this experiment. We train the student PAA-R101 for $1\times$ and $2\times$ schedule.

As reported in Tab. 4, the criteria yield similar results in both $1\times$ and $2\times$ schedule training. Hence, we use the Std/Mean as the default criterion in all later experiments, since it is simpler and less computationally expensive than the Fisher score. For $1\times$ schedule, LAD and CoLAD perform equivalent and improve the baseline by $+0.7$AP.

Table 4: Compare criteria to dynamically switch teacher and student. PAA-R50 pre-trained by $1\times$ schedule on COCO is used as the initial teacher in LAD and CoLAD.

Student	Method	mAP	Improve
PAA-R101 (train $1\times$)	PAA Baseline	42.6	–
	LAD	43.3	+0.7
	CoLAD - Std/Mean	43.3	+0.7
	CoLAD - Fisher	**43.4**	**+0.8**
PAA-R101 (train $2\times$)	PAA Baseline	43.5	–
	CoLAD - Std/Mean	43.9	+0.4
	CoLAD - Fisher	43.9	+0.4

4.2.2 Impact of network's pretrained weights

Assuming no pretrained teacher, we ask whether it is better to train two networks with CoLAD, rather than conventionally training each one separately. We also try to find how much they are improved, if one of them is already pretrained in COCO. To answer these questions, we train the pair PAA-R50 and PAA-R101 for $1\times$ and $2\times$ schedule with different initialization, and report the results in Tab. 5.

As shown in Tab. 5, when no pretrained teacher available (3^{rd} and 6^{th} row), *networks trained with CoLAD outperfom those that were trained independently*. Specifically, there are $+0.9/+0.5$AP improvement in $1\times$ schedule (3^{rd} *vs.* 1^{st} row), and $+0.8/+0.6$AP in $2\times$ schedule (6^{th} *vs.* 2^{nd} row) compared to the baseline PAA-R50/R101, respectively. Although the improvement in PAA-R50 may be expected thanks to co-training with a larger PAA-R101, the fact that the network PAA-R101 also got improved thanks to its smaller partner PAA-R50 is remarkable.

Secondly, when one of the networks is pretrained on COCO (4^{th} and 5^{th} rows), the improvements are better than without pretraining (3^{rd} row), which is expected. However,

Table 5: CoLAD training of PAA-R50 and PAA-R101 detectors, with different pretrained weights. $1\times/2\times$ denotes that the models are trained or pretrained on COCO with $1\times/2\times$ schedule, respectively. (*) denotes comparing with the Baseline trained with $2\times$ schedule.

Method	Networks	Init.	mAP	Improve
Baseline	PAA-R50	$1\times/2\times$	$40.4/41.6$	–
Baseline	PAA-R101	$1\times/2\times$	$42.6/43.5$	–
CoLAD ($1\times$)	PAA-R50	No	41.3	$+0.9$
	PAA-R101	No	43.1	$+0.5$
CoLAD ($1\times$)	PAA-R50	No	41.6	$+1.2$
	PAA-R101	$1\times$	44.1	$+0.6$*
CoLAD ($1\times$)	PAA-R50	$1\times$	42.6	$+1.0$ *
	PAA-R101	No	43.3	$+0.7$
CoLAD ($2\times$)	PAA-R50	No	42.4	$+0.8$ *
	PAA-R101	No	44.1	$+0.6$ *

the improvements are somewhat marginal for the students. In $1\times$ training, CoLAD improves the students PAA-R50 from 41.3 to 41.6AP ($+0.3$) (3rd *vs*. 4th row), and PAA-R101 from 43.1 to 43.3AP($+0.2$) (3rd *vs*. 5th row). In $2\times$ training, the improvements are from 42.4 to 42.6AP($+0.2$) for R50 (6th *vs*. 5th row), while R101 reaches to 44.1AP (6th and 4th row). This demonstrates that *CoLAD can replace LAD when a pretrained teacher is not available.*

What's more interesting is that *"one plus one is better than two"*, *i.e*. a teacher which was trained for $1\times$ schedule, after joining CoLAD for another $1\times$, outperforms the one independently trained for $2\times$ schedule. Concretely, CoLAD improves the teachers PAA-R50 from 41.6 to 42.6AP($+1.0$) (1st *vs*. 5th row), and PAA-R101 from 43.5 to 44.1AP($+0.6$) (2nd *vs*. 4th row). Finally, the experiments above support our hypothesis that the dynamic switching mechanism of CoLAD reduces the risk of local minima trapping, and that is why it outperforms the baseline PAA.

4.2.3 Impact of Teacher-Student's relative model sizes

To understand how the relative gap between the model sizes can affect the student's performance, we compare the results of two model pairs, including (PAA-R18, PAA-R50) and (PAA-R18, PAA-R101) using LAD and CoLAD.

Table 6: Evaluate the teacher-student's relative model sizes. The student PAA-R18 is trained with different teachers. $1\times$ means teacher is pretrained on COCO with $1\times$ schedule.

Teacher	Init.	Method	mAP	Improve
None	–	Baseline PAA	35.8	–
PAA-R50	$1\times$	LAD	36.9	$+1.1$
	No	CoLAD	36.5	$+0.7$
PAA-R101	$1\times$	LAD	36.8	$+1.0$
	No	CoLAD	36.6	$+0.8$

Table 6 reports the experiments. We see that the impact of teachers with different capacity and pretrained weights is negligible and in the typical noise ±0.1AP. This contradicts with the paradox [34] that a student's performance degrades when the gap with its teacher is large. This proves that the discrepancy between teacher and student may not be a weakness of LAD and CoLAD. However, from a different viewpoint, this is also a drawback because a better teacher can't improve the student further. Hence, we should also combine LAD with other distillation techniques in order to take the full advantages of the teacher.

4.3. Comparison with State-of-the-art

We compare our proposed CoLAD with other methods on MS COCO *test-dev* set, especially the recent ones addressing the label assignment for single-stage detectors. Following the previous works [21, 15, 53], we train the model with $2\times$ scheduler, randomly scale the shorter size of the image into the range of 640 and 800. We use the PAA-R50 trained with $3\times$ schedule and multi-scale on COCO as the initial teacher [1]. The teacher was evaluated with 43.3AP and 43.8AP on the *minval* and *test-dev* sets, respectively.

Inspired by [53, 10, 52], we also modify the PAA's head architecture by connecting the classification and localization branches by an auxiliary "objectness" prediction, which is supervised implicitly. However, we use a slightly different implement with [53, 10, 52], which we call Conditional Objectness Prediction (COP). Since COP is not our main contribution, we refer the reader to Appendix A for the motivation and more ablation studies. We report the results of the PAA network amended with COP and trained with CoLAD in Tab. 7.

With R101 backbone, CoLAD achieves 46.0AP, consistently outperforming other recent methods for label assignment, such as OTA [15] (45.3AP), IQDet [33] (45.1AP) and PAA [21] (44.8AP) by large margins on all evaluation metrics. With the larger backbone ResNeXt-64x4d-101, our model can further improve to 47.5AP and surpasses all existing methods. Finally, for the ResNeXt-64x4d-101 with Deformable Convolutional Networks (DCN) [56], CoLAD and OTA both achieve the highest score 49.2AP. Note that, with this DCN backbone, the top 4 methods perform very similar (±0.2AP).

Nevertheless, it is not our intention for LAD/CoLAD to become a replacement for current SOTA methods. Instead, we find it useful to use a simple technique, with a smaller and lower performance teacher ResNet50, to successfully boost the current SOTA results. Indeed, as shown in the next section, by simply replacing ResNet50 with a stronger teacher, we can still improve the results above further.

[1] PAA-R50's pretrained weight is provided by MMDetection https://github.com/open-mmlab/mmdetection/tree/master/configs/paa

Table 7: Compare with state-of-the-art (SOTA) methods (single model trained $2\times$ schedule) on COCO *test-dev* set.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-101 backbone						
FCOS[42]	41.5	60.7	45.0	24.4	44.8	51.6
N.Anchor[23]	41.8	61.1	44.9	23.4	44.9	52.9
F.Anchor[49]	43.1	62.2	46.4	24.5	46.1	54.8
SAPD [54]	43.5	63.6	46.5	24.9	46.8	54.6
MAL[20]	43.6	61.8	47.1	25.0	46.9	55.8
ATSS[48]	43.6	62.1	47.4	26.1	47.0	53.6
GFL[26]	45.0	63.7	48.9	27.2	48.8	54.5
A.Assign[53]	44.5	64.3	48.4	25.9	47.4	55.0
PAA[21]	44.8	63.3	48.7	26.5	48.8	56.3
OTA[15]	45.3	63.5	49.3	26.9	48.8	56.1
IQDet[33]	45.1	63.4	49.3	26.7	48.5	56.6
CoLAD[ours]	**46.0**	**64.4**	**50.6**	**27.9**	**49.9**	**57.3**
ResNeXt-64x4d-101 backbone						
FSAF [55]	42.9	63.8	46.3	26.6	46.2	52.7
FCOS[42]	43.2	62.8	46.6	26.5	46.2	53.3
F.Anchor[49]	44.9	64.3	48.5	26.8	48.3	55.9
SAPD[54]	45.4	65.6	48.9	27.3	48.7	56.8
ATSS[48]	45.6	64.6	49.7	28.5	48.9	55.6
A.Assign[53]	46.5	66.5	50.7	28.3	49.7	56.6
PAA[21]	46.6	65.6	50.8	28.8	50.4	57.9
OTA[15]	47.0	65.8	51.1	29.2	50.4	57.9
IQDet[33]	47.0	65.7	51.1	29.1	50.5	57.9
CoLAD[ours]	**47.5**	**66.4**	**52.1**	**29.8**	**51.0**	**59.1**
ResNeXt-64x4d-101-DCN backbone						
SAPD[54]	47.4	67.4	51.1	28.1	50.3	61.5
ATSS[48]	47.7	66.5	51.9	29.7	50.8	59.4
A.Assign[53]	48.3	67.4	52.7	29.2	51.0	60.3
PAA[21]	49.0	67.8	53.3	30.2	51.3	62.2
OTA[15]	**49.2**	67.6	53.5	30.3	52.5	**62.3**
IQDet[33]	49.0	67.5	53.1	30.0	52.3	62.0
CoLAD[ours]	**49.2**	**68.3**	**54.1**	**30.6**	**52.8**	61.9

4.4. Vision Transformer Backbone

Vision Transformer recently demonstrates its potential to replace convolution networks. Therefore, we conduct experiments to verify the generalization of LAD with Transformer backbones. To this end, we select the Swin-B Transformer [32] and use the teacher PAA-ResNeXt-64x4-101-DCN obtained in the section 4.3. Since CoLAD training with these two networks is extremely heavy, we use LAD but with the dynamic switch instead. We follow exactly the training setting in [32], such as Swin-B backbone is pretrained on ImageNet, multi-scale training, batch-size 16, AdamW optimizer, and $3\times$ schedule.

Table 8: Compare the teacher PAA-ResNeXt101-DCN and the student PAA-Swin-B on COCO *test* set.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Teacher -DCN	49.2	68.3	54.1	30.6	52.8	61.9
Student-Swin	**52.0**	**71.3**	**57.2**	**33.8**	**55.7**	**65.1**

As reported in Tab. 8, the student PAA-Swin-B once again surpasses its teacher with a large margin ($+2.8$AP). Furthermore, on COCO *val* set, it achieves 51.4AP, approaching the heavy Cascade Mask-RCNN (51.9AP) [32] using the same backbone, but without training segmentation. Finally, we recycle the PAA-Swin-B as a teacher to train PAA-R50/R101 with SoLAD, and compare the results to other distillation methods in Tab. 9.

Note that, it is not advisable to judge which method is better based on Tab. 9, since each serves for a particular detector, thus having different baselines. Nevertheless, our SoLAD still achieves the highest performance for the same backbone and training schedule.

Table 9: Using PAA-Swin-B Transformer teacher to train PAA-R50/R101 with SoLAD. † denotes PAA head amended with COP. Results are compared with other methods using the same backbone on COCO *test* set.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet50 backbone - train $1\times$ schedule						
SoLAD	42.9	62.2	46.8	25.1	46.5	54.0
SoLAD †	**43.7**	**63.0**	**48.0**	**25.8**	**47.5**	**54.9**
FGFI [43]	39.9	–	–	22.9	43.6	52.8
TADF [39]	40.1	–	–	23.0	43.6	53.0
DeFeat [17]	40.9	–	–	23.6	44.8	53.0
LD-GF [51]	41.2	58.8	44.7	23.3	44.4	51.1
GID [11]	42.0	60.4	45.5	25.6	45.8	54.2
ResNet101 backbone - train $2\times$ schedule						
SoLAD †	**47.9**	**67.1**	**52.7**	**29.6**	**51.9**	**59.6**
GF2 [25]	46.2	64.3	50.5	27.8	49.9	57.0
LD-GF2 [51]	46.8	64.5	51.1	28.2	50.7	57.8

5. Conclusion

This paper introduced a general concept of label assignment distillation (LAD) for object detection, in which we use a teacher network to assign training labels for a student. Different from all previous distillation methods which force the student to learn directly from the teacher's output, LAD indirectly distills the teacher's experience through the cost of the label assignment, thus defining more accurate training targets. We demonstrate a number of advantages of LAD, notably that it is very simple and effective, flexible to use with most of detectors, and complementary to other distillation techniques. Later, we introduced the Colearning dynamic Label Assignment Distillation (CoLAD) to allow two networks to be trained mutually based on a dynamic switching criterion. Experiments on the challenging MS COCO dataset show that our method significantly surpasses all the recent label assignment methods. We hope that our work on LAD provides a new insight when developing object detectors, and believe that it is a topic worth further exploration.

Acknowledgement

We would like to thank Francisco Menendez, Su Huynh, Vinh Nguyen, Nam Nguyen and other colleagues for valuable discussion and helping reviewing the paper.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[2] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. Tide: A general toolbox for identifying object detection errors. In *ECCV*, 2020.

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: High Quality Object Detection and Instance Segmentation. pages 1–14, 2019.

[5] Yuhang Cao, Kai Chen, Chen Change Loy, and Dahua Lin. Prime sample attention in object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11583–11591, 2020.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12346 LNCS:213–229, 2020.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

[8] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30, 2017.

[9] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[10] Qiang Chen, Yingming Wang, Tong Yang, Xiangyu Zhang, Jian Cheng, and Jian Sun. You only look one-level feature. *arXiv preprint arXiv:2103.09460*, 2021.

[11] Xing Dai, Zeren Jiang, Zhao Wu, Yiping Bao, Zhicheng Wang, Si Liu, and Erjin Zhou. General instance distillation for object detection. *arXiv preprint arXiv:2103.02340*, 2021.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[13] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. *arXiv preprint arXiv:2101.03697*, 2021.

[14] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015.

[15] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. Ota: Optimal transport assignment for object detection. *arXiv preprint arXiv:2103.14259*, 2021.

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[17] Jianyuan Guo, Kai Han, Yunhe Wang, Han Wu, Xinghao Chen, Chunjing Xu, and Chang Xu. Distilling object detectors via decoupled features. *arXiv preprint arXiv:2103.14475*, 2021.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[20] Wei Ke, Tianliang Zhang, Zeyi Huang, Qixiang Ye, Jianzhuang Liu, and Dong Huang. Multiple anchor learning for visual object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 10203–10212, 2020.

[21] Kang Kim and Hee Seok Lee. Probabilistic anchor assignment with iou prediction for object detection. *arXiv preprint arXiv:2007.08103*, 2020.

[22] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

[23] Hengduo Li, Zuxuan Wu, Chen Zhu, Caiming Xiong, Richard Socher, and Larry S Davis. Learning from noisy anchors for one-stage object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10588–10597, 2020.

[24] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 6356–6364, 2017.

[25] Xiang Li, Wenhai Wang, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss v2: Learning reliable localization quality estimation for dense object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11632–11641, 2021.

[26] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *arXiv preprint arXiv:2006.04388*, 2020.

[27] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[30] Benlin Liu, Yongming Rao, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Metadistiller: Network self-boosting via meta-learned top-down distillation. In *European Conference on Computer Vision*, pages 694–709. Springer, 2020.

[31] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[33] Yuchen Ma, Songtao Liu, Zeming Li, and Jian Sun. Iqdet: Instance-wise quality distribution sampling for object detection. *arXiv preprint arXiv:2104.06936*, 2021.

[34] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5191–5198, 2020.

[35] Yongming Rao, Dahua Lin, Jiwen Lu, and Jie Zhou. Learning globally optimized object detector via policy gradient. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6190–6198, 2018.

[36] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:779–788, 2016.

[37] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[39] Ruoyu Sun, Fuhui Tang, Xiaopeng Zhang, Hongkai Xiong, and Qi Tian. Distilling object detectors with task adaptive regularization. *arXiv preprint arXiv:2006.13108*, 2020.

[40] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[41] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. *arXiv preprint arXiv:2003.05664*, 2020.

[42] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9627–9636, 2019.

[43] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4933–4942, 2019.

[44] Shengkai Wu, Xiaoping Li, and Xinggang Wang. Iou-aware single-stage object detector for accurate localization. *Image and Vision Computing*, 97:103911, 2020.

[45] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. *arXiv preprint arXiv:1904.04971*, 2019.

[46] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

[47] Hongkai Zhang, Hong Chang, Bingpeng Ma, Naiyan Wang, and Xilin Chen. Dynamic R-CNN: Towards high quality object detection via dynamic training. In *ECCV*, 2020.

[48] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (2):9756–9765, 2020.

[49] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. Freeanchor: Learning to match anchors for visual object detection. *arXiv preprint arXiv:1909.02466*, 2019.

[50] Xiaosong Zhang, Fang Wan, Chang Liu, Xiangyang Ji, and Qixiang Ye. Learning to match anchors for visual object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[51] Zhaohui Zheng, Rongguang Ye, Ping Wang, Jun Wang, Dongwei Ren, and Wangmeng Zuo. Localization distillation for object detection. *arXiv preprint arXiv:2102.12252*, 2021.

[52] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. *arXiv preprint arXiv:2103.07461*, 2021.

[53] Benjin Zhu, Jianfeng Wang, Zhengkai Jiang, Fuhang Zong, Songtao Liu, Zeming Li, and Jian Sun. Autoassign: Differentiable label assignment for dense object detection. *arXiv preprint arXiv:2007.03496*, 2020.

[54] Chenchen Zhu, Fangyi Chen, Zhiqiang Shen, and Marios Savvides. *Soft Anchor-Point Object Detection*, pages 91–107. 11 2020.

[55] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:840–849, 2019.

[56] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. *arXiv preprint arXiv:1811.11168*, 2018.

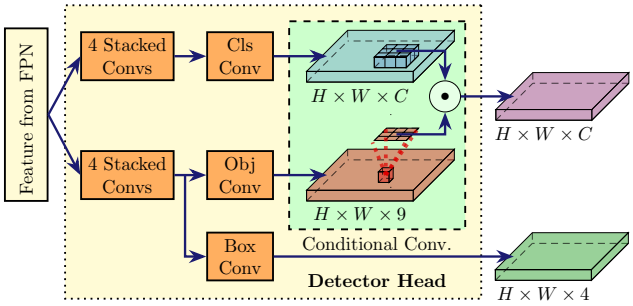


Figure 4: PAA head amended with Conditional Objectness Prediction (COP)

Appendices

A. Conditional Objectness Prediction

A.1. Motivation and Network Architecture

Since object detection performs classification and localization concurrently, their quality must be consistent. For example, a prediction with a high classification probability but low IoU box yields a false positive, while the reverse induces false negative. However, single-stage detectors implement the branches independently, typically each with 4 stacked convolutions. During training and inference, there is no connection between them.

Furthermore, although the two branches have the same computation and feature extraction capacity, the localization receives significantly less training feedback than the classification. This is because most of the samples are negative, which have no box targets for training localization, hence are discarded during gradient backward. Recent methods also add auxiliary branches to predict the localization quality, such as IoU [44, 21], Centerness [25, 49, 42], but is trained only on positive samples. In addition, the positive samples of the same object are often connected and appear in a small local window. However, they are treated independently during non-maxima suppression.

Therefore, we propose adding an auxiliary Conditional Objectness Prediction (COP) to the localization branch. It is similar to the Regional Proposal Network (RPN) of two-stage detector [38] but with renovations, as shown in Fig. 4. Concretely, at each anchor a_i, we predict the objectness scores $\{o_i^k\}_{k=1}^{3\times3}$ of its 3×3 nearest neighbors to capture their spatial correlation. The final classification probability is the dot product of the objectness $\{o_i^k\}$ and the corresponding 3×3 local window of the classification prediction $\{p_i^k\}_{k=1}^{3\times3}$

$$p(a_i) = \frac{1}{9}\sum_{k=1}^{3\times3} o_i^k p_i^k, \tag{4}$$

where o_i and p_i are the confidence score (*i.e.* after Sigmoid) of the objectness branch and classification branch, which are supervised implicitly and mutually through COP product during gradient back-propagation. Therefore, we can fuse and jointly train the branches, make the training consistent with inference. Consequently, all samples in the localization now receive gradient feedback.

Our COP shares a common with the Implicit Object recently introduced in [53, 10], as they are both trained jointly with the classification branch. However, our motivation and implementation are different: (i) We believe features in the regression branch are also helpful to predict objects in the class-agnostic manner, similar to the RPN head in Faster-RCNN, and should not be discarded. COP is introduced to distribute gradient feedback to all samples in the localization branch. (ii) We implement COP as Conditional Convolution [45, 41], where the weights are generated dynamically for each sample. Hence, we can embed the local relationship between the samples to reduce false-positive prediction.

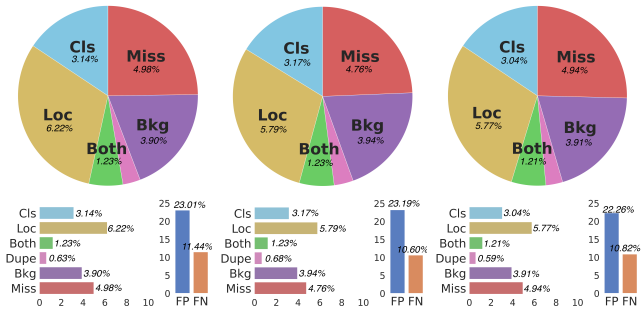
A.2. Ablation Study

We investigate the effectiveness of the Conditional Objectness (COP) with different backbones, including EfficientNet-B0 (Eff-B0) [40], RepVGG-A0 (A0)[13], ResNet18 (R18), and ResNet50 (R50)[18], and compare it with IoU prediction and Implicit Object prediction (IOP). For easy comparison, we use the baseline PAA method, that has IoU prediction by default. Table 10 summarizes the results.

Table 10: Compare different auxiliary predictions: IoU, Implicit Object Prediction (IOP), and Conditional Objectness Prediction(COP) with different backbones. (*) denotes the branch is trained but not used during inference.

Auxiliary Prediction			**mAP**			
IOU	**IOP**	**COP**	**Eff-B0**	**A0**	**R18**	**R50**
✓			32.4	34.0	35.8	40.4
✓	✓		33.4	34.7	36.7	**41.6**
✓		✓	**33.5**	**34.8**	**36.9**	**41.6**
*	✓		33.4	**34.8**	36.7	41.5
*		✓	33.5	**34.8**	**36.9**	**41.6**
	✓		33.3	**34.8**	36.6	41.1
		✓	**33.4**	34.7	**36.9**	**41.2**

At first, we add the IOP or COP to the default PAA head, and observe that both IOP and COP can improve the baseline with considerable margins. For the Eff-B0, A0, R18, R50 backbones, IOP increases $+1, +0.7, +0.9, +1.2$AP, and COP increases $+1.1, +0.8, +1.1, +1.2$AP, respectively. COP and IOP perform equally on R50, but COP



(a) PAA (b) CoLAD (c) CoLAD-COP

Figure 5: Error analysis using TIDE [2] toolbox of the models PAA, CoLAD, and CoLAD-COP with the same backbone ResNet50 on the MS COCO *minval* set.

is slightly better for small backbones Eff-B0($+0.1$AP), A0 ($+0.1$AP), and R18 ($+0.2$AP).

Secondly, we try dropping the IoU prediction during inference and use only IOP or COP, and observe that the results remain almost unchanged (4^{th} and 5^{th} rows).

However, when we train the models without the IoU branch, the performance is dropped more severely for ResNet50 backbone (6^{th} and 7^{th} rows). This proves that IoU is still helpful as deep supervised signal for the regression branch in training, but can be safely omitted during infer.

B. Prediction Error Analysis

Beyond evaluating the mAP metric, we use the TIDE [2] toolbox to analyze the prediction errors of the three models, PAA, CoLAD, and CoLAD-COP, with the same backbone ResNet50.

As shown in Fig. 5, CoLAD and CoLAD-COP help reduce the localization error of the baseline PAA from 6.22% to 5.79% and 5.77%, respectively. CoLAD also reduces the classification error by 3.04%. These indicate that the dynamic mechanism in CoLAD is effective to guide the network to learn a good label assignment, which results in low classification and localization error. In addition, CoLAD can recall more objects, since the false negative percentage is reduced from 11.5% for PAA to 10.85% for both CoLAD and CoLAD-COP. Finally, the introduction of COP can better suppress noisy prediction, as the false positive ratio is reduced from 23.01% to 22.26%.

C. Compare with other distillation methods

Head-to-head comparison of distillation methods for object detection is not easy, since each method is typically developed for a particular detector. Therefore, for reference purpose only, we select LD [51] to compare, since it is based on the SOTA single-stage detector Generalized Focal (GF) [26, 25], which is inline with us but has higher performance than our PAA baseline. However, we emphasize that the two methods address different problems. LD[51] focuses on localization distillation and is applied particularly for GFL detector, while we address the label assignment. Therefore, the two methods can be combined.

Table 11: Compare our LAD techniques to Localization Distillation (LD) [51] for different ResNet backbones. T and S denote teacher and student networks. LAD is based on PAA[21] and LD is based on GF[26]. The results are compared for student networks w.r.t. its baseline on COCO *test* set.

T	S	PAA	LAD	CoLAD	SoLAD	GF	LD
R50	R18	35.8	36.9	36.5	38	36.0	36.1
R101	R18	35.8	36.8	36.6	38.4	35.8	36.5
R101	R50	40.4	41.6	41.3	42.4	40.1	41.1

Table 11 compare the two methods using the same teacher and student's backbones. It is obvious that LAD and CoLAD are superior to LD in all cases. Moreover, our LAD/CoLAD is very simple and can be adapted quickly to any single-stage detectors without architecture modification, and not restricted to Generalized Focal detector [26, 25]. This shows how flexible and effective our method is compared to other distillation methods, such as feature mimicking.